Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
IMD0036 – Sistemas Operacionais

Gerência de Arquivos: Parte 1

Prof. Gustavo Girão
girao@imd.ufrn.br

O que vamos estudar?

- Elementos básicos que compõem um Sistema Operacional

Gerência de Processos

Gerência de Arquivos

Gerência de Memória

Gerência de Entrada/Saída

O que vamos estudar?

- Elementos básicos que compõem um Sistema Operacional

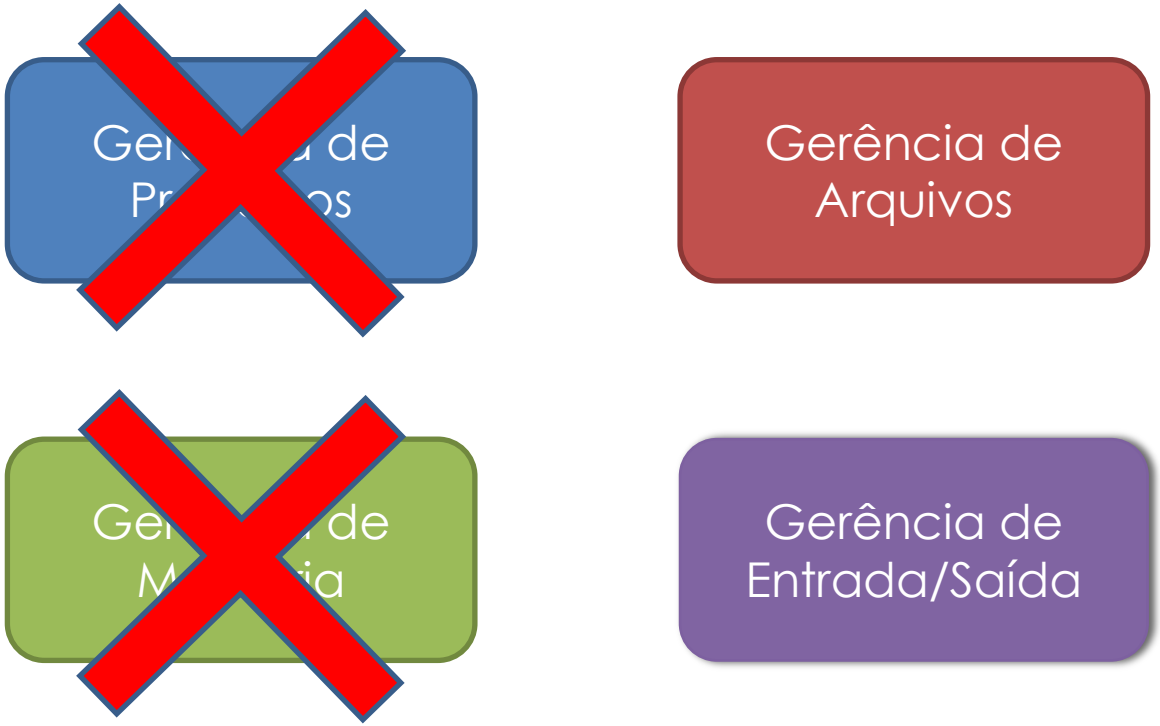
Gerência de Arquivos

Gerência de Memória

Gerência de Entrada/Saída

O que vamos estudar?

- Elementos básicos que compõem um Sistema Operacional



O que vamos estudar?

- Elementos básicos que compõem um Sistema Operacional

Roteiro

- Arquivos
  - Estrutura
  - Atributos
  - Operações
- Diretórios
  - Nível ùnico
  - Hierárquicos
- Implementação
  - Esquemas
  - Arquivos e Diretórios
  - Log
  - Journaling
  - VFS

Introdução

- O sistema de arquivos é a parte mais visivel do SO
- Cria um recurso lógico a partir de recursos fisicos através de uma interface coerente, simples e fácil de usar
- Mecanismos de armazenamento de dados e acesso a programas
- Do ponto de vista da interface, apresenta dois componentes básicos:
  - Arquivos
    - ✧ Armazenamento de dados e programas
  - Diretórios
    - ✧ Organização e informações de arquivos

Introdução

- Arquivos (e diretórios) são corriqueiramente armazenados em Discos
  - A informação precisa ser persistente
  - Precisa ser segura (consistente)
  - Rápida de acessar , gerenciar
- Questões principais:
  - Como encontrar a informação?
  - Como impedir que um usuário tenha acesso a informações de outro usuário?
  - Como saber quais blocos estão livres?

Objetivo

- Fornecer mecanismos para o usuário manipular arquivos e diretórios
- Garantir a validade e coerencia dos dados
  - Minimizar ou eliminar o risco de perda / alteração das informações
- Otimizar o acesso
- Fornecer suporte a outros sistemas de arquivos
- Suporte a varios usuarios (multiprogramação)
  - Uso compartilhado (proteção e acesso concorrente)

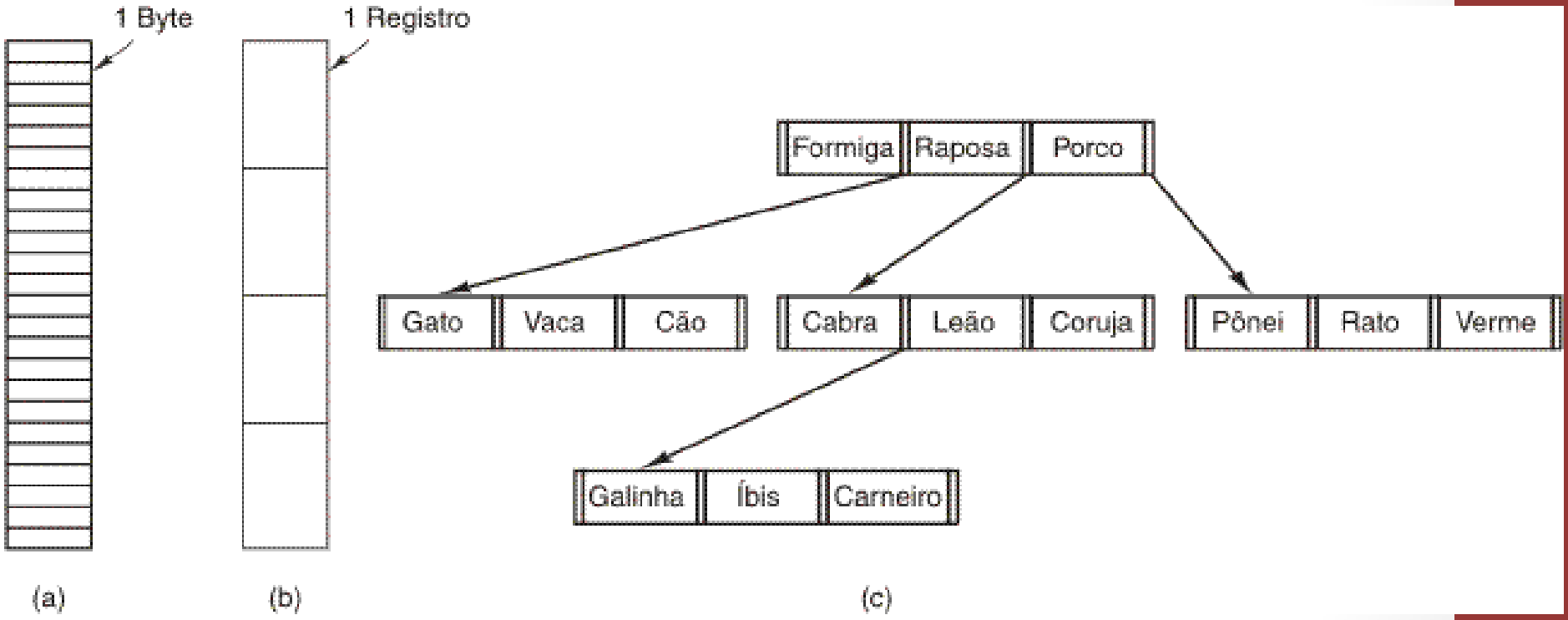
Estrutura de Arquivos

- Como um mecanismo de abstração, arquivos são nomeados de acordo com a decisão de porjeto do sistema operacional.
  - Tipicamente: nome+extensão
- Tamanhos de nomes permitidos podem variar
- O uso de extensões ajudam a identificar o tipo de arquivo e/ou aplicações de usuário associadas a ele
  - Mesmo quando não é necessário. Ex: Linux

Estrutura de Arquivos

- Podem ser estruturados de três maneiras distintas:
  - **Sequências de bytes**
    - Apenas um conjunto desestruturado de bits
    - O sistema operacional não sabe o que é o arquivo.
    - Utilizado por Unix e Windows
  - **Sequências de registros**
    - O arquivos é um conjunto estruturado de registros
    - Cada registro tem um tamanho fixo
    - Leitura: retorna um registro
    - Escrita: adiciona ou sobrepõe um registro
  - **Árvores**
    - Estruturas de gistros organizados na forma de árvore
    - Tamanhos variados
    - Contém um campo-chave que indexa o registro

Estrutura de Arquivos



Tipos de Arquivos

- Arquivos regulares
  - Agregam informações que o usuário gostaria de armazenar
- Diretórios
  - Arquivos de gerência para manter a estrutura do sistema de arquivos
- Arquivos especiais de caracteres
  - Relacionado a entrada e saída de periféricos baseados em caracteres. Ex: teclados, impressoras
- Arquivos especiais de blocos
  - Relacionado a entrada e saída de periféricos baseados em blocos. Ex: discos.

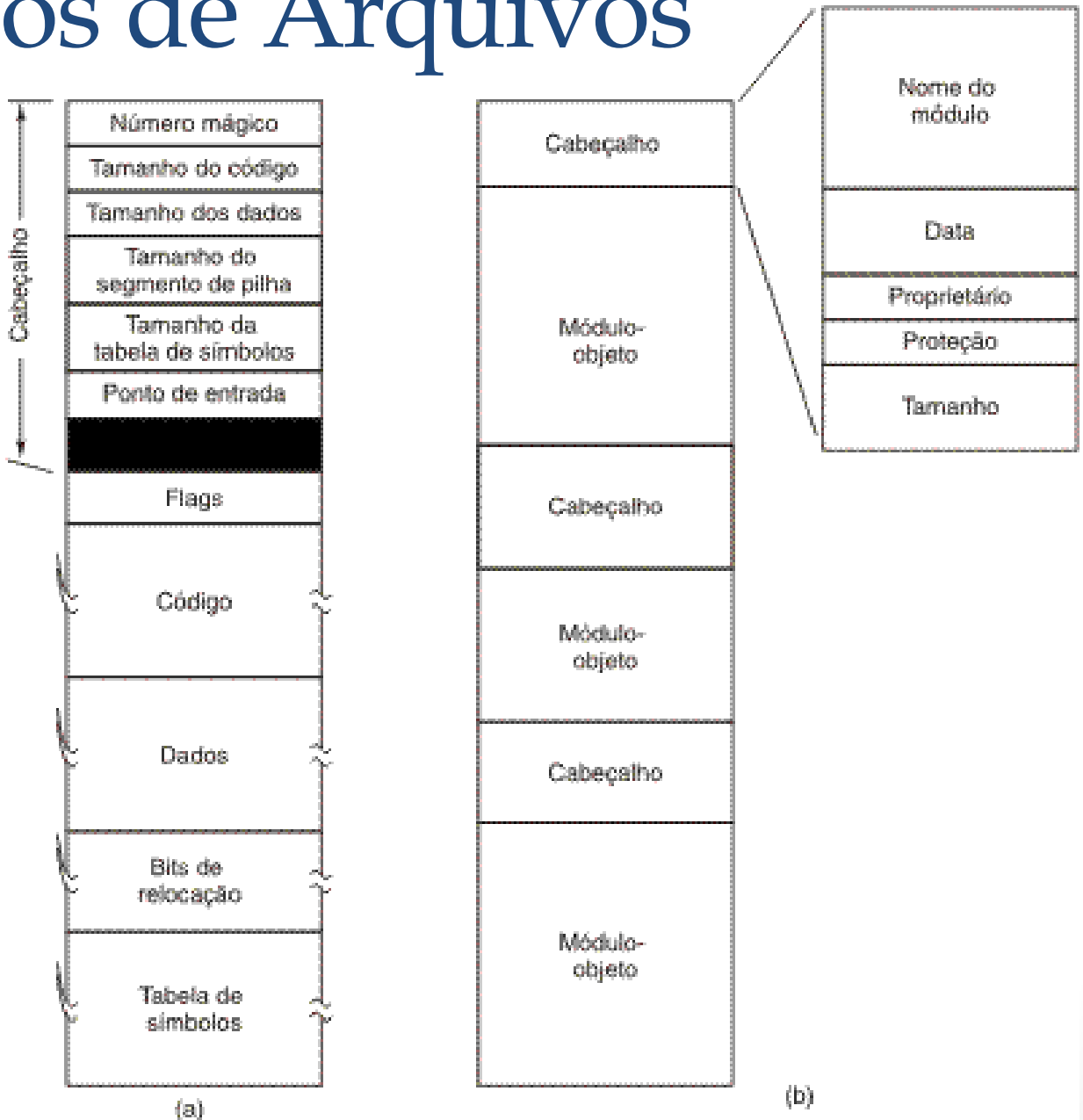
Tipos de Arquivos

- De modo geral, arquivos podem ser ASCII ou binários
  - **Arquivos ASCII** representam uma formatação legível aos seres humanos e são uma representação direta da informação
    - Contém um cabeçalho com informações de data de criação, proprietário, etc
  - **Arquivos binários** são uma representação específica dependente da aplicação
    - Contém também outras informações relevantes como:
      - Número mágico
      - Tamanho do texto (código)
      - Tamanho dos dados
      - Tamanho dos dados não –inicializados
      - Tamanho da tabela de símbolos
      - Ponto de entrada (por onde começar a executar?)

Acesso a arquivos

- Acesso sequencial
  - Utilizados nos primeiros sistemas operacionais
  - Basicamente em função das tecnologias de memória secundária existentes (fitas magnéticas, cartões, etc)
- Acesso aleatório
  - Necessário para aplicações que respondem a eventos aperiódicos. Ex.: Acesso a bancos de dados diversos, controle de entrada/saída de pessoas

Tipos de Arquivos



Atributos de Arquivos

- Arquivos contém informações de nome e dados.
- Porém são necessários outros tipos de informações que também estão associadas aos arquivos:
  - **Proteção**
  - **Flags de controle**
  - **Registro por chave**
  - **Atributos de momento e tamanho**

	Atributo	Significado
Proteção	Proteção	Quem pode ter acesso ao arquivo e de que maneira
	Senha	Senha necessária para ter acesso ao arquivo
	Criador	ID da pessoa que criou o arquivo
	Proprietário	Atual proprietário
Flags de Controle	Flag de apenas para leitura	0 para leitura/escrita; 1 se apenas para leitura
	Flag de oculto	0 para normal; 1 para não exibir nas listagens
	Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
	Flag de repositório (*archive*)	0 se foi feita cópia de segurança; 1 se precisar fazer cópia de segurança
	Flag ASCII/binário	0 para arquivo ASCII; 1 para arquivo binário
	Flag de acesso aleatório	0 se apenas para acesso seqüencial; 1 para acesso aleatório
	Flag de temporário	0 para normal; 1 para remover o arquivo na saída do processo
	Flag de impedimento	0 para desimpedido; diferente de zero para impedido
Registros por chave	Tamanho do registro	Número de bytes em um registro
	Posição da chave	Deslocamento da chave dentro de cada registro
	Tamanho da chave	Número de bytes no campo-chave
Atributos de momento e tamanho	Momento da criação	Data e horário em que o arquivo foi criado
	Momento do último acesso	Data e horário do último acesso ao arquivo
	Momento da última mudança	Data e horário da última mudança ocorrida no arquivo
	Tamanho atual	Número de bytes no arquivo
	Tamanho máximo	Número de bytes que o arquivo pode vir a ter

Operações

- Várias chamadas de sistema podem ser usadas para manipular arquivos:
  - Create
  - Delete
  - Open
  - Close
  - Read
  - Write
  - Append
  - Seek
  - Get attributes
  - Set Attributes
  - Rename

Operações

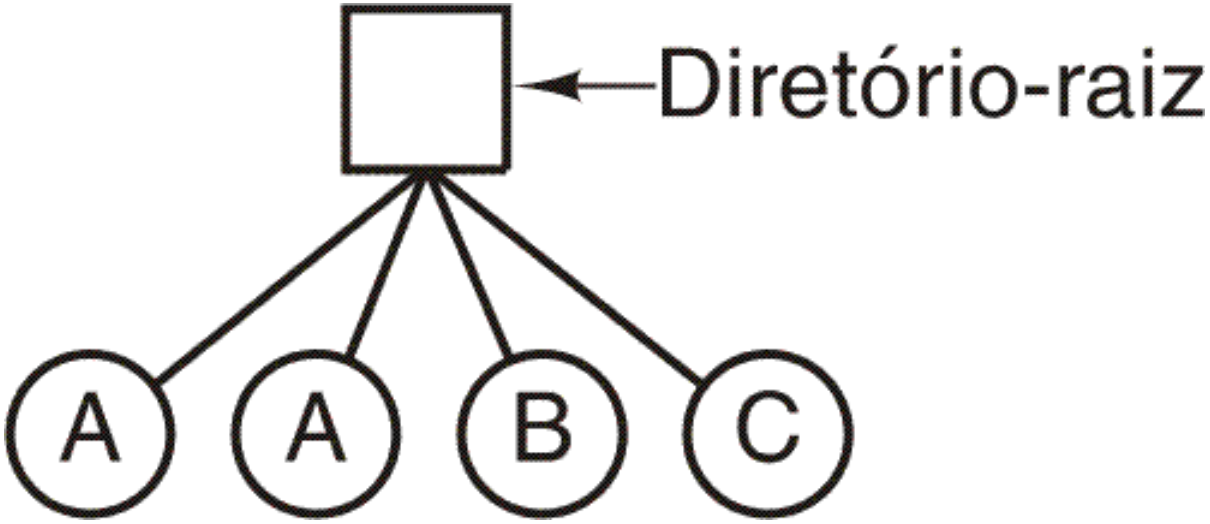
- A manipulação dos arquivos se dá através de descritores de arquivos
  - Identificadores únicos dos arquivos dentro da biblioteca do sistema operacional

Diretórios

- Operações realizadas num diretório
  - Busca
  - Criação
  - Remoção
  - listar conteúdo
- Estruturas de diretório
  - organizam os arquivos presentes no sistema
  - Maneira mais simples: baseando-se em um **diretório-raiz**

Diretórios

- **Diretório em nível único**
  - Utilizado em sistemas operacionais antigos e sistemas muito simples
  - Funciona bem quando se trata de um sistema monousuário

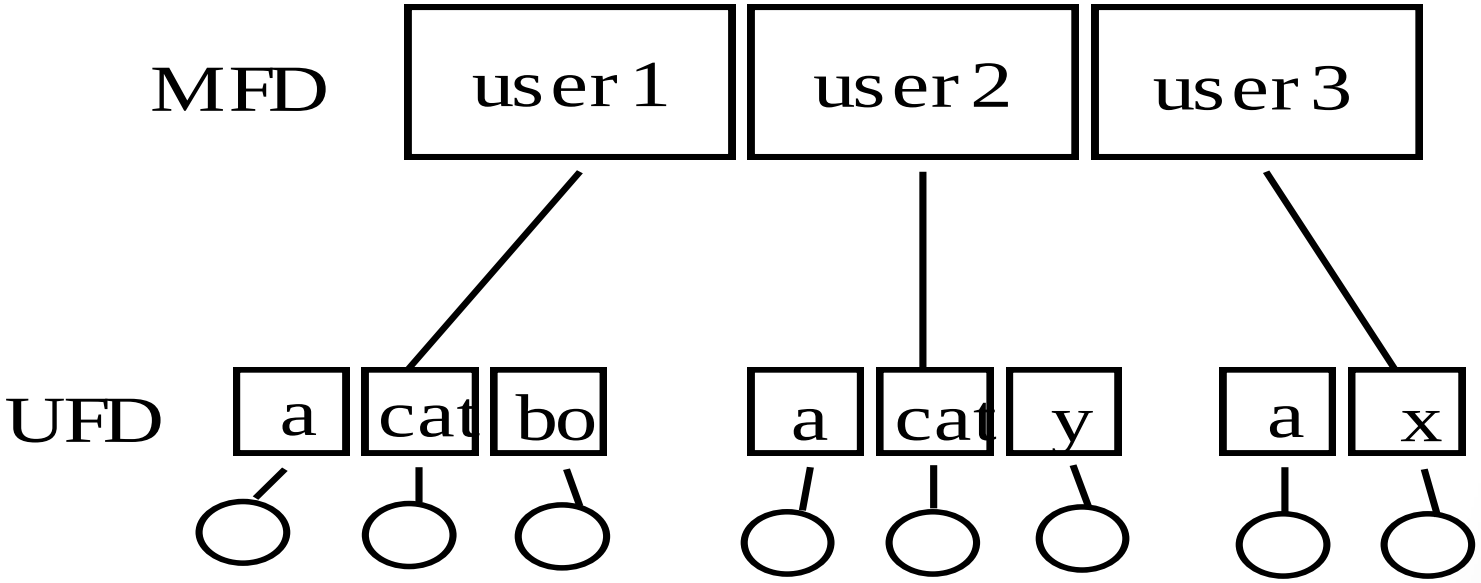


Diretórios

- Diretórios hierárquicos
  - Cada usuário tem seu próprio diretório de arquivo de usuário (UFD)
    - Cada diretório de usuário tem uma estrutura similar
  - Diretório de arquivo mestre (MFD)
    - Isola usuários
  - Problema: compartilhamento de arquivos, se for permitido
    - Cria-se uma ligação simbólica, porém o arquivo tem apenas um proprietário
      - O que acontece se um usuário resolve excluir o arquivo?
      - Duplicar a informação?

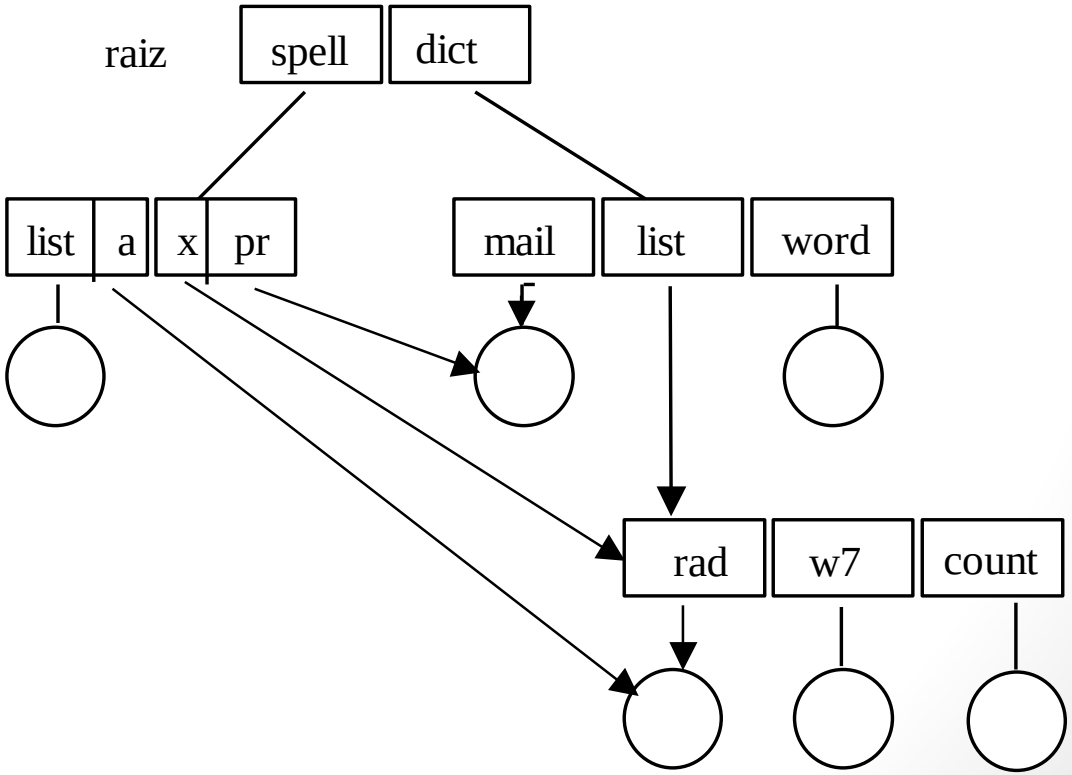
Diretórios

- Diretórios hierárquicos



Diretórios

- Diretório em grafo acíclico
  - Permite compartilhamento de arquivos e subdiretórios
  - Generalização do esquema de diretório estruturado em árvore



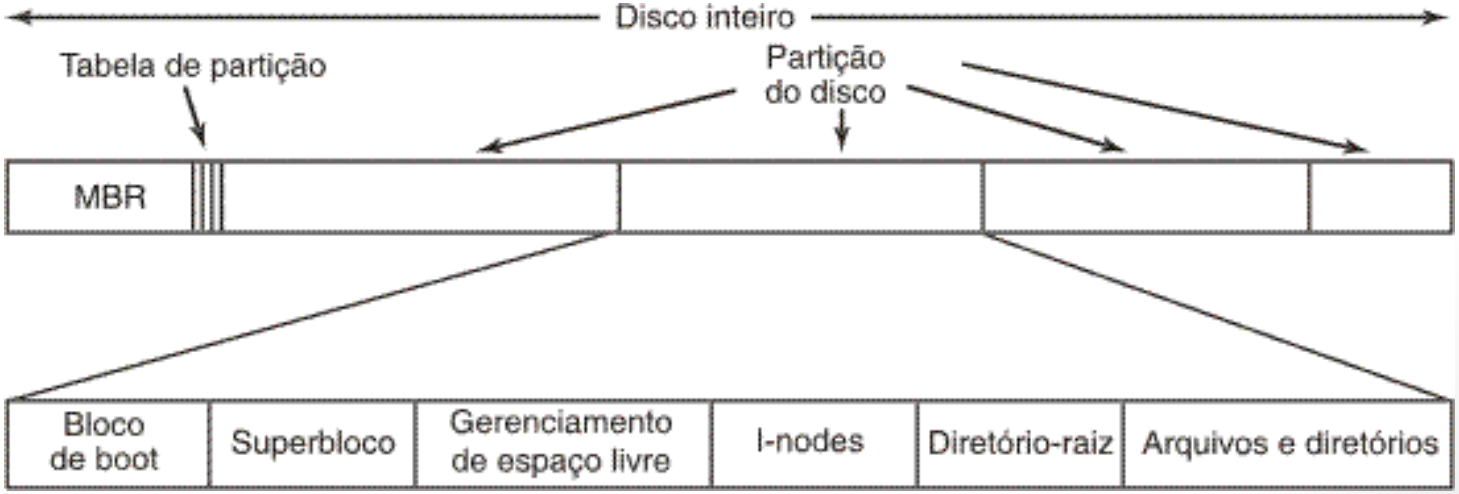
IMPLEMENTAÇÃO DO SISTEMA DE ARQUIVOS

Esquema do sistema de arquivos

- Os próprios sistemas de arquivos (conjunto de estruturas de dados para gerênciamento) são armazenados em discos
  - MBR (master boot record): registro mestre de inicialização
    - Inicializa o computador
  - Tabela de partição
    - Endereços iniciais e finais da partição
    - Cada partição contém informações relevantes para o sistema de gerenciamento de arquivos
  - O conteúdo do arquivo em si é armazenado nos **blocos** dos discos
    - **Bloco**: unidade básica de um disco
    - Fragmentação interna!

Esquema do sistema de arquivos

- Partição
  - Bloco de inicialização
    - ✧ Um código que carrega um sistema operacional presente na partição
  - Superbloco
    - ✧ Contém parametros de projeto do sistema de arquivos (estrutura dos arquivos, tamanho de registros, etc)
- As informações contidas na partição dependem do mecanismo de gerencia de arquivos adotado

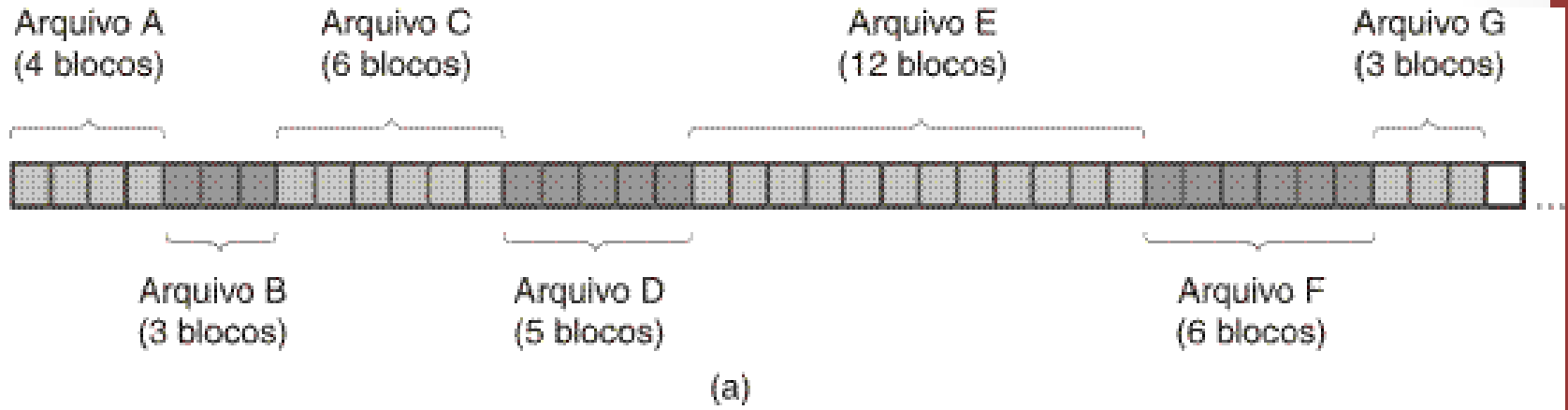


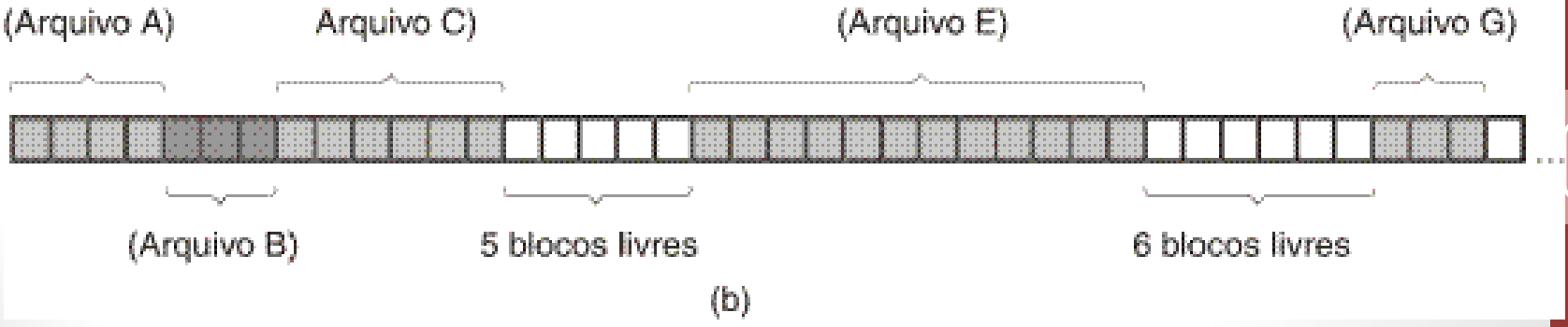
Manutenção e controle de blocos

- Alocação contígua
  - Os bytes do arquivo são armazenados em blocos contíguos do disco
  - Implementação simples pois o controle sobre os blocos é reduzido a:
    - O endereço em disco do primeiro bloco
    - Número de blocos do arquivo
  - Ótimo desempenho no acesso:
    - Uma vez encontrado o bloco inicial, só é necessário ler o numero de blocos seguintes que o arquivo ocupa.
  - Problema:
    - Fragmentação externa!

Manutenção e controle de blocos

- Alocação contígua



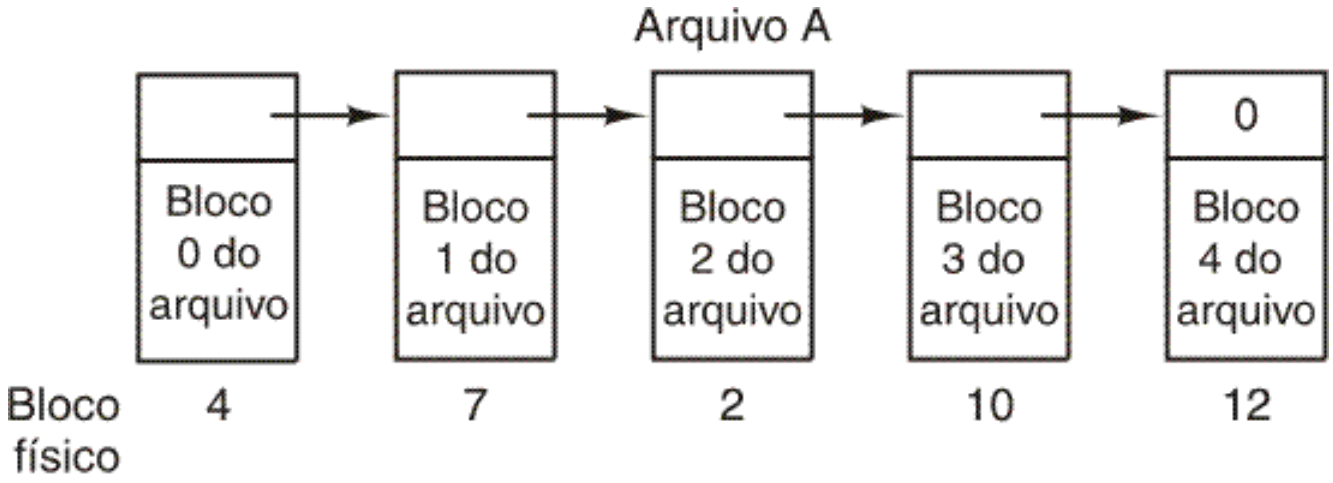


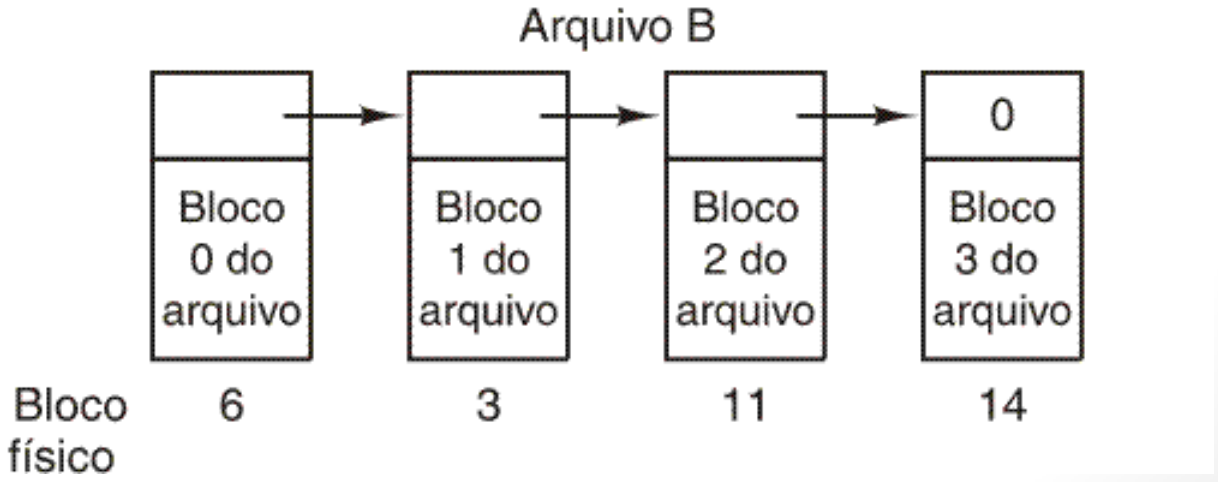
Manutenção e controle de blocos

- Alocação por lista encadeada
  - Todo bloco pode ser utlizado (não precisa ser em sequência.
  - Uma estrutura de dados orienta qual bloco é o próximo na sequencia do arquivo
  - Vantagens
    - não há fragmentação externa
    - tamanho dos arquivos pode ser mudado facilmente
  - Desvantagens
    - acesso aleatório é mais demorado
    - maior fragilidade em caso de problemas
  - Base de funcionamento da **FAT** (File Allocation Table)
    - Sistema de arquivos em versões antigas do Windows

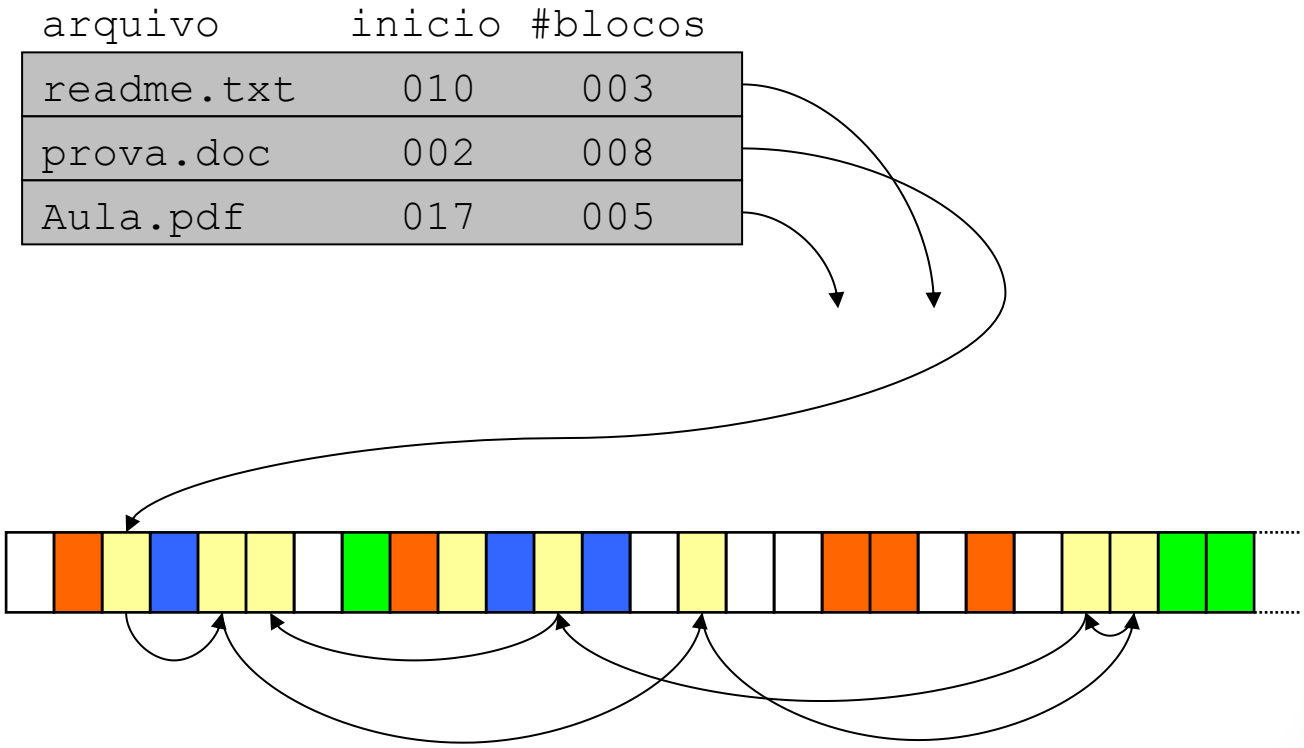
Manutenção e controle de blocos

- Alocação por lista encadeada



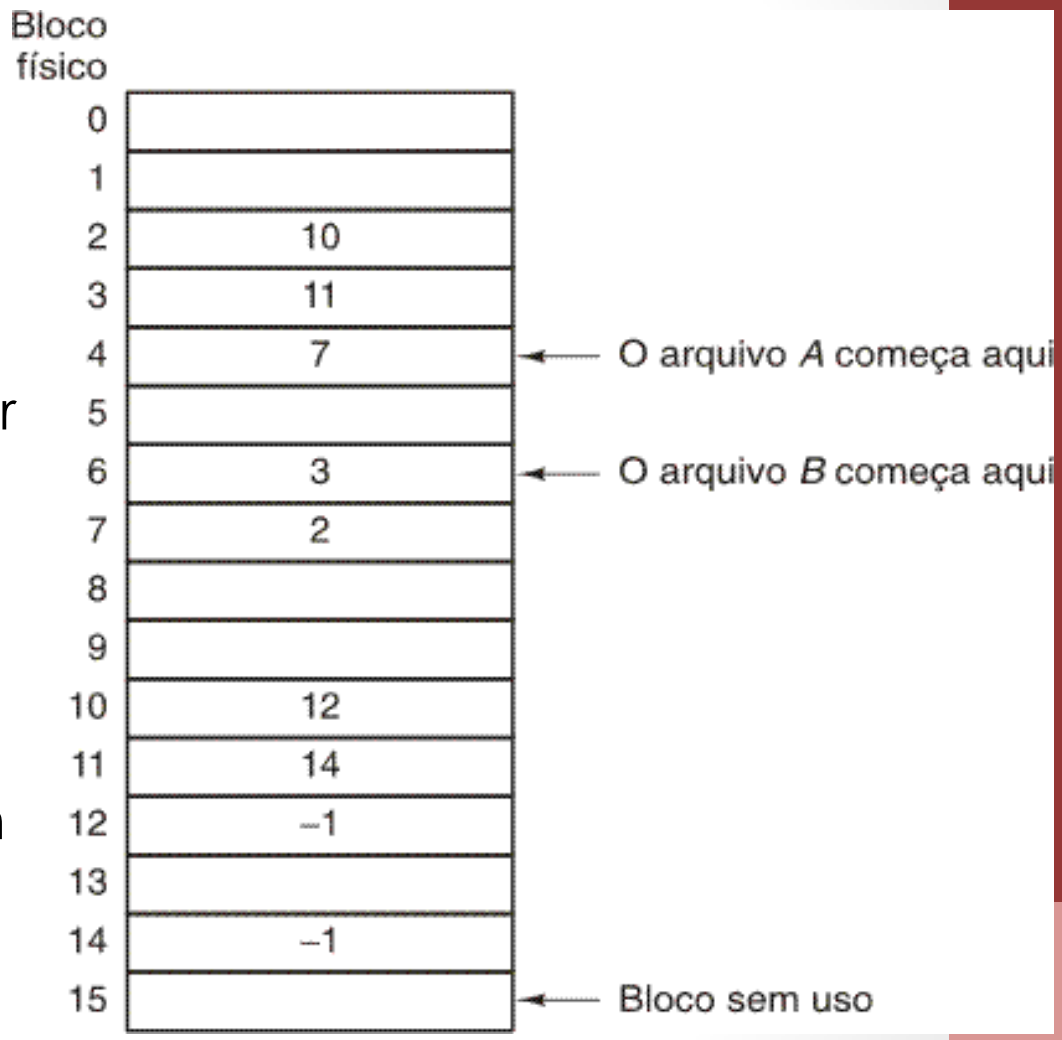


Alocação encadeada



Manutenção e controle de blocos

- Alocação por lista encadeada
  - O acesso aleatório pode se rmuito lento. Uma solução é manter uma tabela de blocos na memória
  - O arquivo tem um bloco inicial e aponta para o próximo
  - O arquivo finaliza com um valor padrão (-1, por exemplo)

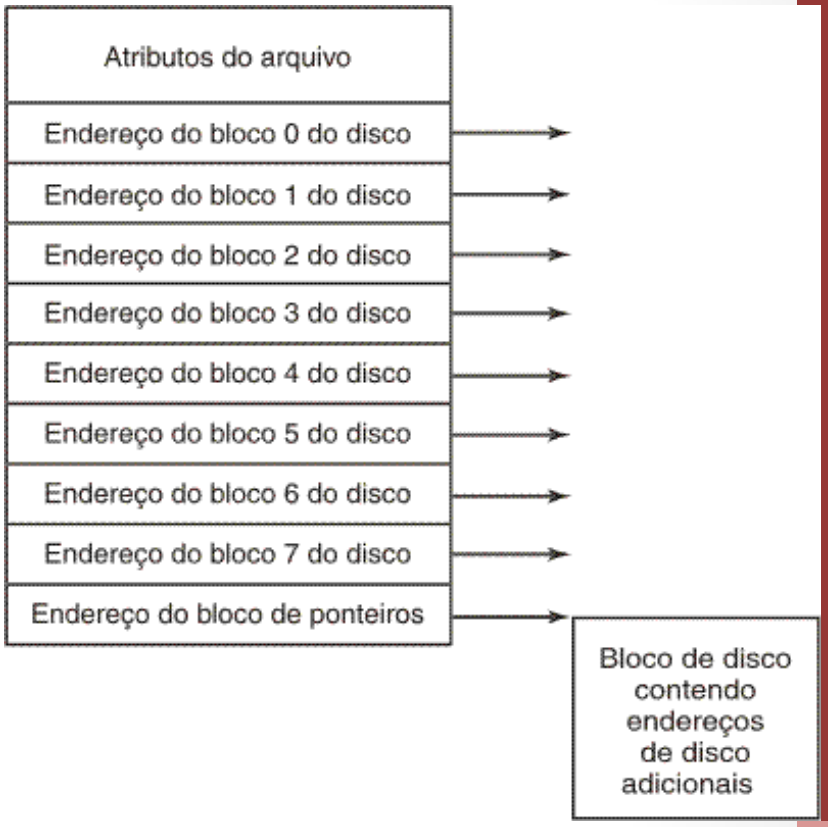


Alocação indexada

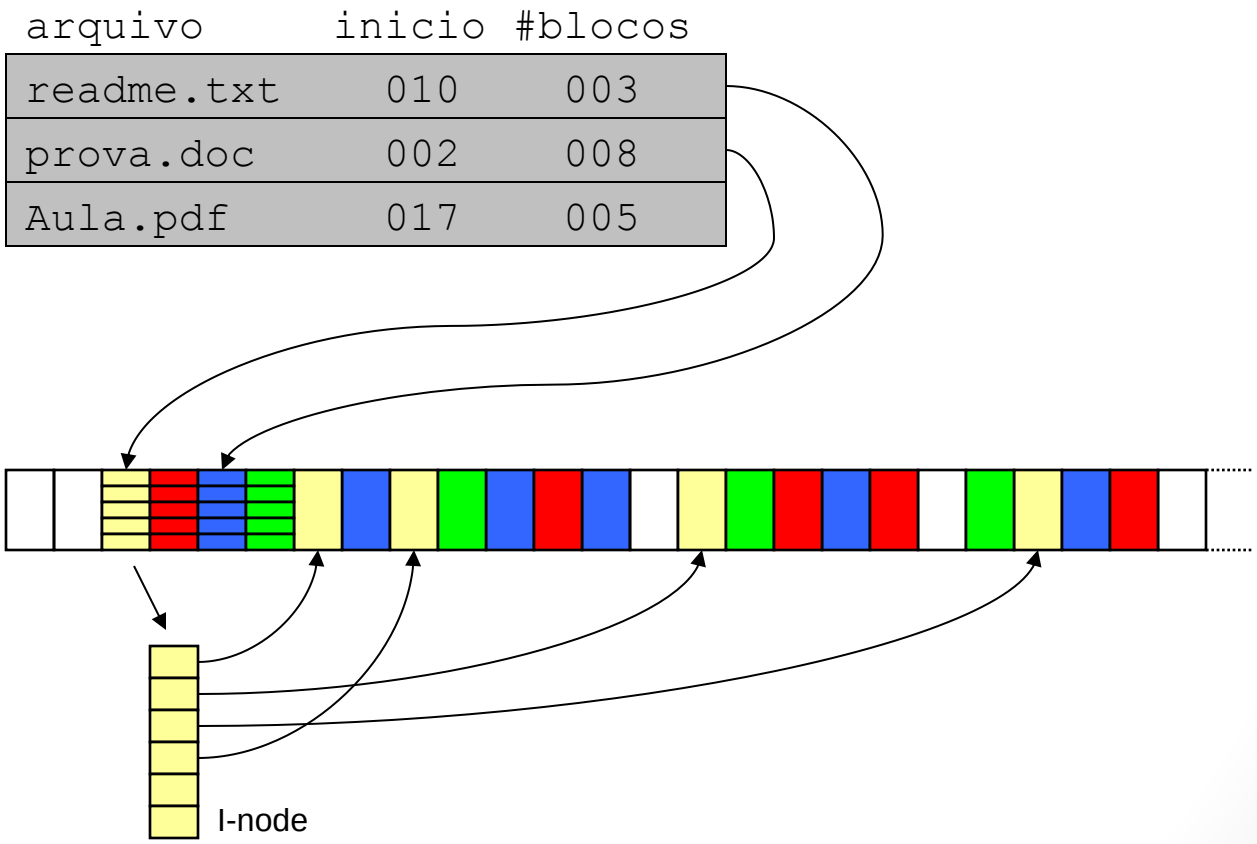
- Baseada em tabelas de blocos
  - um bloco especial guarda a tabela de blocos do arquivo: **index-node** (**i-node**)
  - diretório aponta para os i-nodes
  - blocos podem estar espalhados
  - O conjunto de i-nodes podem estar em uso ou livres para serem associados a um arquivo
- Base de funcionamento do UNIX
- Vantagem sobre listas encadeadas em memória:
  - As informações do arquivo só precisam ser carregadas quando o arquivo estiver aberto
  - Cria uma restrição quanto ao número de arquivos abertos

Alocação indexada

- Para dar suporte ao crescimento do arquivo pode-se utilizar ponteiros:
  - O i-node pode ser dividido em partes
- Contém informações sobre:
  - permissões do arquivo
  - contagem de links
  - blocos duplicados
  - blocos ruins
  - associações de tamanho e
  - ponteiros para os blocos de dados.



Alocação indexada

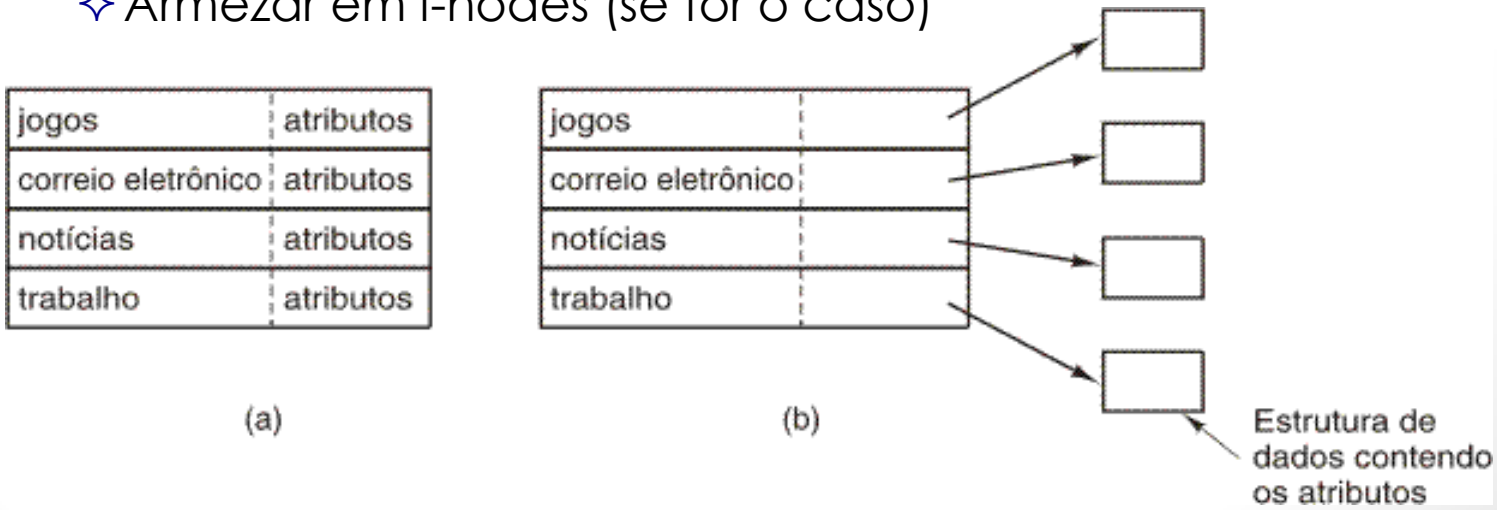


Alocação indexada

- Vantagens
  - não há fragmentação externa
  - todo o disco pode ser usado
  - acesso rápido
  - robustez em caso de problemas
- Desvantagens
  - gerência mais complexa
  - espaço em disco perdido com os i-nodes

Implementação de Diretórios

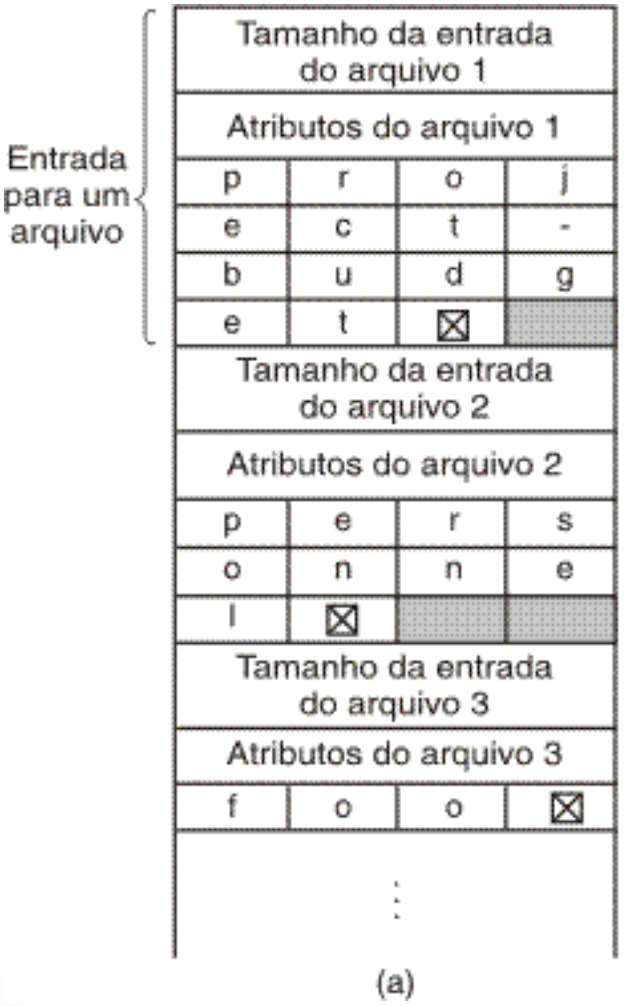
- Com a implementação de um sistema de diretórios, este é responsável por fornecer informações dos arquivos contidos nele
  - Estes atributos dizem respeito a data de criação, permissões, proprietário, etc.
  - Onde armazenar tais atributos?
    - Na entrada do diretório
    - Armezar em i-nodes (se for o caso)

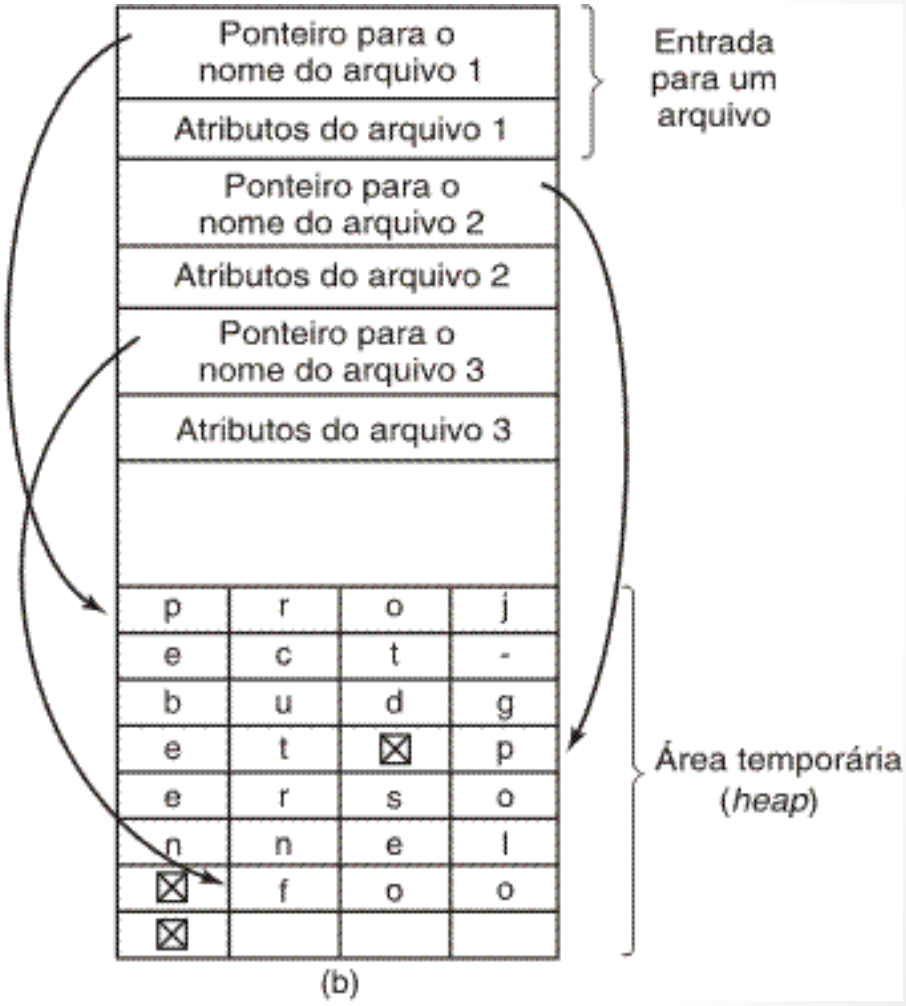


Implementação de Diretórios

- Nomes de arquivos podem ter tamanhos variados
  - Esta informação precisa constar após os atributos do arquivo
  - Esta informação pode estar organizada por arquivo ou unificando todas as informações para diminuir a perda de espaço
    - Em ambos os casos, utiliza-se um caracter terminador
  - A busca por um nome de arquivo, em ambos os casos pode ser
    - **Linear**: simples mas lenta
    - **Utilizando uma tabela hash**: mais rápida mas mais complexa

Implementação de Diretórios





Sistemas baseados em log

- Para tentar minimizar acessos ao disco, novos sistemas de arquivos implementam tecnologias baseadas em log:
  - A idéia é registrar na memória as últimas escritas feitas em arquivos (um log de alterações)
  - Periodicamente essas alterações são escritas no fim do disco criando um log das últimas alterações
    - ✧ Escritas contíguas são mais rápidas, por isso é melhor do que atualizar diretamente cada arquivo no disco
    - ✧ Mas e o tamanho do log??
      - Escritas seguidas podem liberar espaço para minimizar o impacto do tamanho do log i.e.: no primeiro período o arquivo ***foo*** foi modificado e registrado no log. Pouco tempo depois um novo registro de modificação de ***foo*** chega no disco. Basta armazenar a última alteração

Journaling

- Como manter a confiabilidade ao modificar um arquivo?
- Considere qeu para remover um arquivo são necessários 3 passos:
  - Remover o arquivo de seu diretório
  - Liberar o i-node para o conjunto de i-nodes livres
  - Voltar todos os blocos do arquivo para o conjunto de blocos livres do disco
- Se o sistema não realizar as 3 operações **atomicamente**, haverá inconsistência

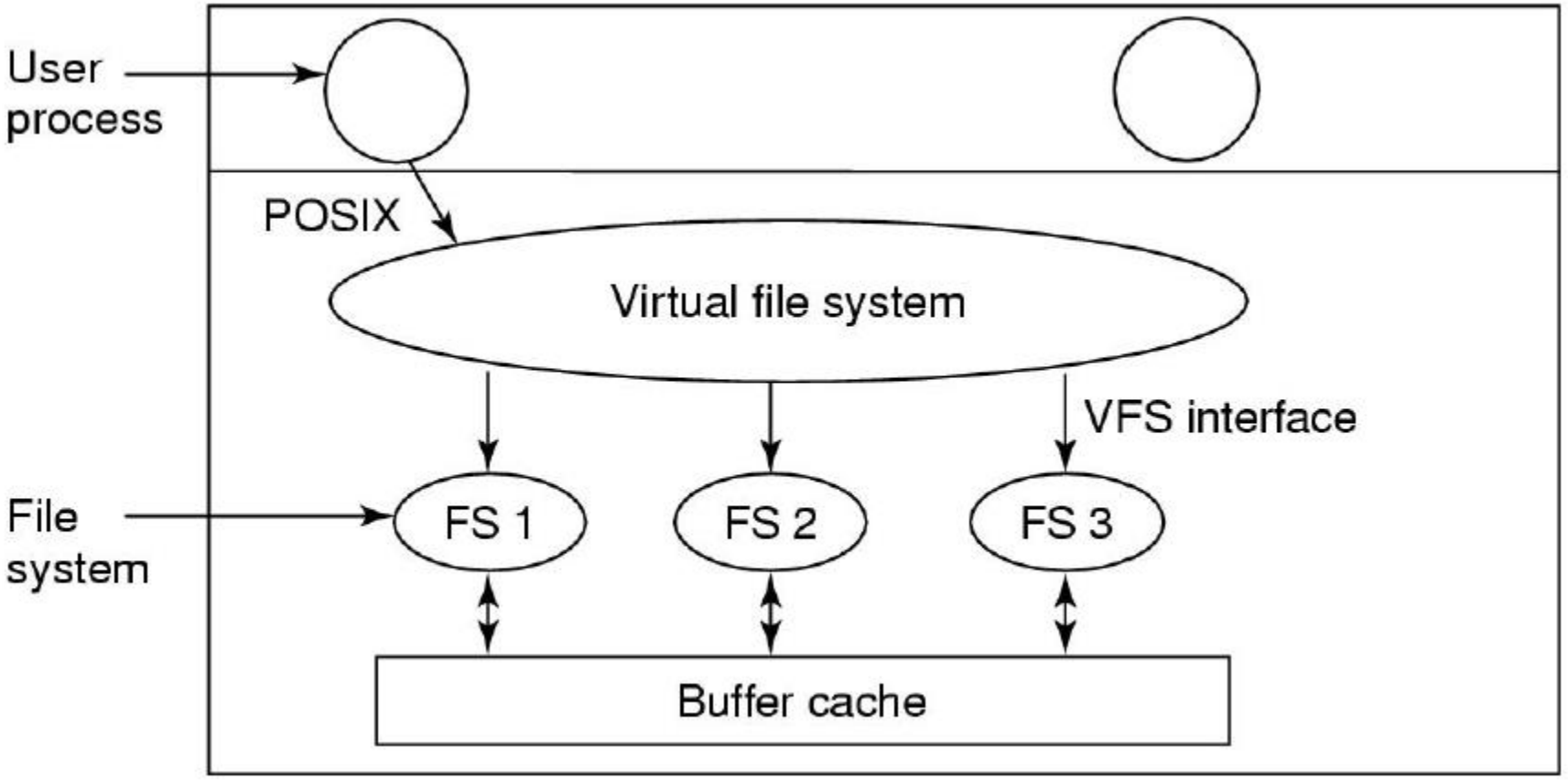
Journaling

- É um tipo de sistema de arquivos de log
  - Cada atualização no sistema de arquivos é armazenada em um arquivo de log (transação)
    - Uma transação é considerada aceita quando, quando escrita no arquivo de log
    - O sistema de arquivo não necessariamente esta atualizado
  - As transações no arquivo de log são processadas de forma assíncrona
    - Quando a transação é processado, a mesma é armazenada no arquivo de log
  - Se um sistema é reinicializado, as transações no arquivo são processadas antes do início da utilização do sistema
  - Duas escritas são necessárias: log + sistema de arquivos

Sistemas de Arquivos Virtuais

- Como integrar diferentes sistemas de arquivos?
  - Um disco mais antigo que contenha arquivos FAT-16
- Virtual File System
  - Uma camaa intermediária recebe a solicitação de acesso feita pelo usuário (chamada de sistema)
  - Este sistema reconduz a solicitação para o sistema de arquivos necessário fazendo uso de informações do **superbloco**

Sistemas de Arquivos Virtuais



Próxima Aula

- Otimização de Sistemas de Arquivos
- Discos e RAID

Referências

- TANENBAUM, Andrew S.. **Sistemas operacionais modernos**. 3. ed. São Paulo: Prentice Hall, 2009. 653 p. ISBN: 9788576052371.
  - **Capítulo 4**

- OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo. **Sistemas operacionais**. 4. ed. Porto Alegre: Bookman, 2010. ISBN: 9788577805211.
  - **Capitulo 8**

- SILBERCHATZ, A.; Galvin, P.; Gagne, G.; **Fundamentos de Sistemas Operacionais**, LTC, 2015. ISBN: 9788521629399
  - **Capítulo 9**